Proceedings of the 2006 IEEE
International Conference on Mechatronics and Automation
June 25 - 28, 2006, Luoyang, China

Optimal Base Placement for a Discretely Actuated Robotic Manipulator (D-ARM)

MIYAHARA, Keizo
Department of Mechanical Engineering
Graduate School of Engineering
Osaka University
Suita, Osaka 565-0871, JAPAN
Email: miyahara@mech.eng.osaka-u.ac.jp

Gregory S. Chirikjian
Department of Mechanical Engineering
G.W.C.Whiting School of Engineering
Johns Hopkins University
Baltimore, Maryland 21218, USA
Email: gregc@jhu.edu

***Abstract*— A “Discretely Actuated Robotic Manipulator”, or “D-ARM”, is any member of a class of robotic manipulators powered by actuators that have only discrete positional stable states such as solenoids. One of the most significant kinematic phenomena of D-ARMs is the discreteness of both input range and end-effector frames. The main characteristics of D-ARMs are: stability at each state without feedback loop; high task repeatability; mechanism simplicity; minimal supporting devices; low cost. These are strong advantages for all engineering field, including: manufacturing automation; mobile robot; space structure; micro/nano mechanism.**

This paper illustrates a pre-processing step that conditions the baseline manipulator for its kinematic synthesis. The computation derives the optimal pose of the baseline manipulator for minimal alteration that will meet the design criteria. The proposed process utilizes $SE(3)$-differentiation techniques. The conducted simulations demonstrate the feasibility of the synthesis method.

I. Introduction

Actuators can be recognized as belonging to one of the following two kinematic categories: “Continuous-Range-of-Motion Actuator (Continuous Actuator)”, or “Discrete-Range-of-Motion Actuator (Discrete Actuator)” [1]. The former one is continuously position controllable and accepts a continuous range of input command values, as ordinary servo motors do. The other has only a finite number of discrete stable positions, and thus the input command has also a discrete nature.

We define a class of manipulators called the “Discretely Actuated Robotic Manipulator (D-ARM)” or “Discrete Arm” which is powered by discrete actuators. In particular, a “Binary Actuated Robotic Manipulator (B-ARM)” or “Binary Arm” is one with actuators that have only binary stable states. Further, in contrast to D-ARM, let us call a manipulator a “Continuously Actuated Robotic Manipulator (C-ARM)”, or “Continuous Arm”, if the manipulator uses continuous actuators, as most standard robotic systems do. Table I summarizes this categorization.

TABLE I
Actuator and Manipulator Categorization

Actuator	Example	Stable state = Input command	Manipulator
Continuous	Servomotor	Continuous range	C-ARM
Discrete	Solenoid, Pneumatic cylinder	Discrete range	D-ARM

A fundamental example is shown in Fig. 1. The 2D (two dimensional) B-ARM that has three bi-stable actuators is activated according to a three-digit binary number. The left (right) actuator is associated with the most (least) significant bit of the binary number, and the center actuator corresponds to the middle one. The binary bit “1” (“0”) means full extension (contraction) of the actuator. By changing the binary number given to the controller, one of eight ($= 2^3$) possible frames (elements of $SE(3)$, the Special Euclidean group in three space) of the end-effector can be reached with a certain binary number by the 3-bit B-ARM . In Fig. 2, 68.7 billion ($= 64^6$) end-frames are “reachable” by the 3D B-ARM example that consists of six Stewart-type binary platforms. The discussion above is also applicable for the s-state (multi-state) general type of D-ARM in a similar manner [2].

One of the most significant kinematic phenomena of D-ARMs is the discreteness of both input range and end-effector frames as shown above.

While discrete actuators are the key components of D-ARMs, they are also widely used as stand-alone motion sources in various applications. The variety of those applications originates from the following significant characteristics of discrete actuators:

(1) Stability at each state without feedback loop
(2) High task repeatability
(3) Mechanism simplicity, including kinematic parameter adjustment (e.g., stroke stopper for a pneumatic cylinder)
(4) Minimal supporting devices, especially feedback systems
(5) Low cost and small volume due to (3) and (4)

All of these characteristics of discrete actuators directly result in those of a D-ARM which is composed of discrete actuators. These characteristics are strong advantages for many engineering fields, including: manufacturing automation; mobile robotics; space structures; micro/nano mechanisms. One

1-4244-0466-5/06/$20.00 ©2006 IEEE

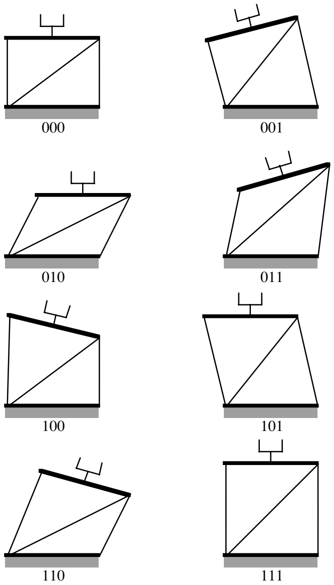


Fig. 1. An Example D-ARM (3-bit 2D Binary Parallel Platform)

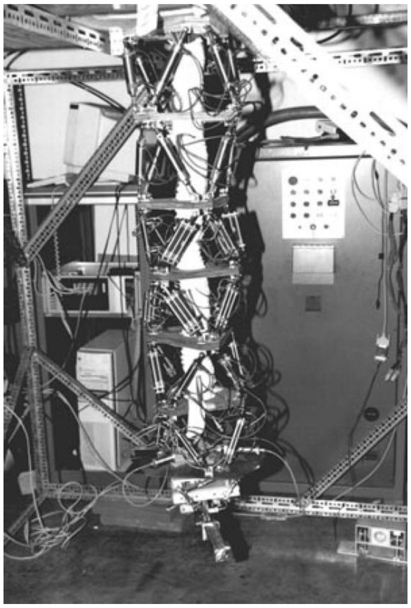


Fig. 2. An Example D-ARM (6 × 6-bit 3D Binary Parallel/Serial Hybrid Arm)

of the most recent attempts to utilize the advantages of the discrete actuation can be found on the researches for robotic planetary explorers [3]. The mobile robot employs binary actuators to achieve a light weight and durable system with a simple and thus reliable controller.

To control robot motion continuously, such as a trajectory constrained task, continuous actuators are essential devices for the manipulator system. It should be noticed, however, that continuous actuation is not so essential for the teaching-playback work especially if the task is an ordinary pick-and-place one, since continuity of input range is not necessary for the playback phase. The key issue is the fact that, after each teaching process, the robot's actuators only go back and forth between several memorized "discrete" positions. Namely, continuous actuators are used as the robot's hardware in order to adapt a task by software programming for such a teaching phase.

Reconsidering continuous actuators from the point of view of cost efficiency, they seem to be "overkill" [4] when a task defines start/goal end-effector frames but the trajectory is less important as long as it is bounded. Conversely, discrete actuators could be a sufficient and cost effective solution for such a playback task if they have the ability to easily change kinematic parameters, such as stroke length for a cylinder.

One of the most fundamental synthesis issues for manipulator design is to determine its kinematic parameters in order for reaching all the given desired frames. Considering the kinematic synthesis discretely actuated mechanical system with higher degrees of freedom (DOF), several studies have been performed [2], [4]–[6]. The proposed synthesis processes in the works are aim to solve the following fundamental and important inverse kinematic problem of a D-ARM:

> Given a D-ARM (base-line design) and finite sets of desired positions and orientations of the D-ARM's end-effector (desired frames: $\mathbf{H}_{ee} \in SE(N)$, where $N = 2(3)$ for 2D(3D) case), determine kinematic parameters (the vector $\mathbf{a}$) of the manipulator.

This paper addresses another problem in D-ARM synthesis to reach a desired set of end-effector poses, i.e., we discuss the issue of how to place a D-ARM in space so as to minimize the amount of change required in its kinematic parameters to reach a set of desired frames. This issue is substantially different from the problem addressed in earlier works because here we do not seek to optimize over actuator stop lengths. Rather, the computations in this paper can be viewed as a pre-processing step that conditions the baseline design so that minimal alteration will meet the design criteria.

II. Gradient for functions on $SE(3)$

The common concept of optimization over $\mathbb{R}^N$ based on the derivative operation is the following:

> Given a function $f : \mathbb{R}^N \to \mathbb{R}$, we search for $\mathbf{x} \in \mathbb{R}^N$ such that $f(\mathbf{x})$ is a minimum by first picking a value of $\mathbf{x}$ and computing the gradient $\nabla f = [\partial f/\partial x_1, ..., \partial f/\partial x_N]^T$.

This gradient computation may be performed analytically or numerically. Updating the current value of $\mathbf{x}$ is performed numerically by moving a small amount in the direction opposite to ∇f. This process is iterated until $\nabla f = \mathbf{0}$, in which case the corresponding value of $\mathbf{x}$ is a local minimum. In some cases it is possible to solve the equation $\nabla f = \mathbf{0}$ analytically for all possible minima, but in general this is difficult to do.

In a similar manner, the optimization method can be derived as shown below in which the concept of differentiation on the

Lie group $SE(3)$ is used. The following formulation provides the analytical foundations for the remainder of this paper:

> Suppose G is the Lie group under consideration (which for the sake of discussion here can be either $SO(3)$ or $SE(3)$) and $\mathbf{g} \in G$. If there is a function $f : G \to \mathbb{R}$, we seek the value of $\mathbf{g}$ such that $f(\mathbf{g})$ is minimized.

One can build a gradient descent algorithm in the context when G is a Lie group in almost the same way as for $\mathbb{R}^N$ as follows. First, define partial derivatives. This is done by perturbing the current value of $\mathbf{g}$ by an infinitesimal motion along the basis directions $\{\mathbf{X}_1, ..., \mathbf{X}_n\}$ of the Lie group, where n is the dimension of the group. The resulting equation is as follows [7]:

$$X_i^R f = \frac{d}{dt} f(\mathbf{g} \circ \exp(t\mathbf{X}_i))|_{t=0}$$

or

$$X_i^L f = \frac{d}{dt} f(\exp(t\mathbf{X}_i) \circ \mathbf{g})|_{t=0} \,. \tag{1}$$

These are respectively called the right and left derivatives of f along the $\mathbf{X}_i$ direction in the Lie algebra of G at the group element $\mathbf{g}$. In general $X_i^R f \neq X_i^L f$, so we need to choose if we will do a "left gradient descent" or a "right gradient descent." In the current context it does not matter which one is used.

After the values $X_i^R f$ (or $X_i^L f$) for $i = 1, ..., n$ are computed, the current value of $\mathbf{g}$ can be updated as

$$\mathbf{g} \leftarrow \mathbf{g} \circ \exp\left(-\epsilon \sum_{i=1}^{n} X_i^R f \cdot \mathbf{X}_i\right)$$

or

$$\mathbf{g} \leftarrow \exp\left(-\epsilon \sum_{i=1}^{n} X_i^L f \cdot \mathbf{X}_i\right) \circ \mathbf{g}, \tag{2}$$

where ϵ is a small positive real number. The new value of $\mathbf{g}$ is substituted into $f(\cdot)$, and the process of computing the gradient and updating iterates until $f(\mathbf{g})$ is minimized. Just as in the case of $\mathbb{R}^N$, the negative gradient direction is used because it is the direction of steepest descent in the local neighborhood of $\mathbf{g}$.

It is possible, just as in $\mathbb{R}^N$, that for some functions $f(\cdot)$, setting the gradient to zero from the beginning can lead to analytical expressions that can be solved for the desired minimal value of $\mathbf{g} \in G$. This has been done for three examples in [8] using a slightly different formulation. One of these examples is revisited here using (1). This is the example of the RMS superposition of two sets of points in $\mathbb{R}^3$. This is a problem that has been studied extensively in the computer vision/imaging literature [9] , crystallography [10], and spacecraft attitude estimation [11]. However, it is usually approached by using Lagrange multipliers as in [12] instead of using the intrinsic group-theoretic approach presented here.

Consider two sets of points in $\mathbb{R}^3$, each of which contains m elements: $\{\bar{\mathbf{z}}_1, \ldots, \bar{\mathbf{z}}_m\}$ and $\{\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_m\}$. We seek the rigid-body motion $\mathbf{g} = (\mathbf{R}, \bar{\mathbf{b}})$ such that the first set of points is optimally superimposed on the second. In order to achieve this goal, the following mean-square-error function must be minimized:

$$f(\mathbf{g}) = \sum_{i=1}^{m} w_i ||\bar{\mathbf{y}}_i - \mathbf{g} \circ \bar{\mathbf{z}}_i||^2. \tag{3}$$

Here $||\cdot||$ is the two-norm representing the Euclidean distance between the i^{th} point in one set and the rigidly moved version of the i^{th} point in the other, $w_i > 0$ is a weighting value often chosen to be the number one, and

$$RMS = \min_{\mathbf{g} \in G} \sqrt{f(\mathbf{g})} \tag{4}$$

is the weighted root-mean-squared error between the two point sets.

Taking the left gradient of f, we see that

$$\begin{aligned} X_k^L f &= \frac{d}{dt} \sum_{i=1}^{m} w_i ||\bar{\mathbf{y}}_i - (\mathbf{I} + t\mathbf{X}_k) \circ \mathbf{g} \circ \bar{\mathbf{z}}_i||^2|_{t=0} \\ &= 2 \sum_{i=1}^{m} w_i (\bar{\mathbf{y}}_i - \mathbf{g} \circ \bar{\mathbf{z}}_i) \cdot (-\mathbf{X}_k(\mathbf{g} \circ \bar{\mathbf{z}}_i)) \,. \end{aligned} \tag{5}$$

In the case of spatial rigid-body motion, we can write for $k = 1, ..., 6$ all of the resulting scalar equations as one vector equation, and set

$$\sum_{i=1}^{m} w_i (\bar{\mathbf{y}}_i - \mathbf{g} \circ \bar{\mathbf{z}}_i)^T \mathbf{S} (\mathbf{g} \circ \bar{\mathbf{z}}_i) = 0 \,, \tag{6}$$

where $\mathbf{S} = \sum_{i=1}^{6} s_i \mathbf{X}_i$ and s_i are undetermined parameters. Equation (6) must hold for all possible values of $\mathbf{S}$, hence the first step in finding $\mathbf{g}$ is to rewrite this equation so as to isolate $\mathbf{S}$ where in the current context $\mathbf{S}$ is of the form

$$\mathbf{S} = \begin{pmatrix} \mathbf{\Omega} & \bar{\mathbf{v}} \\ \mathbf{0}^T & 0 \end{pmatrix}, \tag{7}$$

where $vec(\mathbf{\Omega}) = \bar{\boldsymbol{\omega}}$. Making this substitution into (6) and using the fact that $\bar{\boldsymbol{\omega}}$ and $\bar{\mathbf{v}}$ are arbitrary, we get the following equation and condition [8]:

$$\sum_{i=1}^{m} w_i (\bar{\mathbf{y}}_i - \mathbf{g} \circ \bar{\mathbf{z}}_i)^T \begin{pmatrix} -\mathbf{R}\mathbf{Z}_i\mathbf{R}^T - \mathbf{B} & \mathbf{I} \\ \mathbf{0}^T & 0 \end{pmatrix} = \mathbf{0} \tag{8}$$

$$\bar{\mathbf{b}}_{opt} = \frac{\sum_{i=1}^{m} w_i (\bar{\mathbf{y}}_i - \mathbf{R}_{opt} \bar{\mathbf{z}}_i)}{\sum_{i=1}^{m} w_i} \,, \tag{9}$$

where $\bar{\mathbf{b}} = \boldsymbol{vec}(\mathbf{B})$, $\bar{\mathbf{z}}_i = \boldsymbol{vec}(\mathbf{Z}_i)$, $\mathbf{I}$ is the 3×3 identity matrix and $\mathbf{R}_{opt}$ is the proper orthogonal matrix in the polar decomposition of the matrix:

$$\mathbf{P} = \sum_{i=1}^{m} w_i \bar{\mathbf{y}}_i \bar{\mathbf{z}}_i^T \,. \tag{10}$$

That is, $\mathbf{R}_{opt} = (\mathbf{P}\mathbf{P}^T)^{-\frac{1}{2}}\mathbf{P}$ and the optimal pose is $\mathbf{g}_{opt} = (\mathbf{R}_{opt}, \bar{\mathbf{b}}_{opt})$.

This result is the same answer as obtained when using Lagrange multipliers, though we have avoided the introduction of those unnecessary additional variables.

III. Optimal Base Frame Placement Problem

We consider the following problem related to D-ARM synthesis:

> Given a D-ARM that has already been designed, at what pose should the base be fixed in order for the end-effector to be able to reach a set of discrete desired poses?

As we shall see, this problem has a closed-form solution, and can be formulated using the terminology of motion metrics. To begin, call the poses that are desirable to be reached $\mathbf{h}_1,...,\mathbf{h}_{n_{frm}}$. Define a subset of the poses of the current design that are reachable relative to the D-ARM's base as $\mathbf{g}_1,...,\mathbf{g}_{n_{frm}}$. We choose this subset to have the same number of elements as the number of desired frames, and assign a correspondence.

A. Optimal Placement by Gradient Descent

The current restricted problem of D-ARM placement is equivalent to finding the base pose $\mathbf{b} \in SE(3)$ such that the cost function

$$C(\mathbf{b}) = \sum_{i=1}^{n_{frm}} d^2(\mathbf{b} \circ \mathbf{g}_i, \mathbf{h}_i) \tag{11}$$

is minimized where $d(\cdot,\cdot)$ is a metric on $SE(3)$. The solution to the problem of finding the optimal $\mathbf{b} \in SE(3)$ can be solved numerically as a gradient descent on $SE(3)$. This problem can be solved in closed-form using the same solution as for the RMS positioning problem when we take the metric on rigid-body displacement [13] with $\mathbf{W} = \alpha\mathbf{I}$ where $\mathbf{I}$ is the identity matrix and $w_{44} = 1$.

If $\mathbf{b} = (\mathbf{B}, \bar{\mathbf{b}})$, $\mathbf{g}_i = (\mathbf{R}_i, \bar{\mathbf{r}}_i)$ and $\mathbf{h}_i = (\mathbf{A}_i, \bar{\mathbf{a}}_i)$, the cost function can be expanded as

$$C(\mathbf{B}, \bar{\mathbf{b}}) = \sum_{i=1}^{n_{frm}} \left\{\alpha||\mathbf{B}\mathbf{R}_i - \mathbf{A}_i||^2 + ||\mathbf{B}\bar{\mathbf{r}}_i + \bar{\mathbf{b}} - \bar{\mathbf{a}}_i||^2\right\} , \tag{12}$$

where $||\cdot||$ is used to denote both the matrix and vector two-norm, and it is clear from the argument which is being used. The matrix two-norm is defined for any $\mathbf{M} \in \mathbb{R}^{n\times n}$ as

$$||\mathbf{M}|| = \sqrt{\mathrm{trace}\,(\mathbf{M}\mathbf{M}^T)}\,. \tag{13}$$

The optimal values of $\mathbf{B}$ and $\bar{\mathbf{b}}$ can be found in closed-form by rewriting Equation (12) in a form similar to the RMS position problem solved earlier. To observe this, consider the following notation:

$$\begin{aligned} \mathbf{A}_i &= [\bar{\mathbf{a}}_i^{(1)}, \bar{\mathbf{a}}_i^{(2)}, \bar{\mathbf{a}}_i^{(3)}]\,; \\ \mathbf{R}_i &= [\bar{\mathbf{r}}_i^{(1)}, \bar{\mathbf{r}}_i^{(2)}, \bar{\mathbf{r}}_i^{(3)}]\,. \end{aligned} \tag{14}$$

Equation (12) can be written as

$$C(\mathbf{B}, \bar{\mathbf{b}}) = \sum_{i=1}^{n_{frm}} \left\{\alpha\left(\sum_{k=1}^{3}||\mathbf{B}\bar{\mathbf{r}}_i^{(k)} - \bar{\mathbf{a}}_i^{(k)}||^2\right) + ||\mathbf{B}\bar{\mathbf{r}}_i + \bar{\mathbf{b}} - \bar{\mathbf{a}}_i||^2\right\} , \tag{15}$$

where now all the norm signs represent vector two-norms. By defining

$$\bar{\mathbf{r}}_i^{(0)} = \bar{\mathbf{r}}_i - \frac{1}{n_{frm}}\sum_{i=1}^{n_{frm}} \bar{\mathbf{r}}_i \tag{16}$$

and

$$\bar{\mathbf{a}}_i^{(0)} = \bar{\mathbf{a}}_i - \frac{1}{n_{frm}}\sum_{i=1}^{n_{frm}} \bar{\mathbf{a}}_i\,, \tag{17}$$

$\bar{\mathbf{b}}$ can be taken to be zero, and Equation (15) is written as

$$C(\mathbf{B}, \mathbf{0}) = \sum_{i=1}^{n_{frm}} \left\{\alpha\left(\sum_{k=1}^{3}||\mathbf{B}\bar{\mathbf{r}}_i^{(k)} - \bar{\mathbf{a}}_i^{(k)}||^2\right) + ||\mathbf{B}\bar{\mathbf{r}}_i^{(0)} - \bar{\mathbf{a}}_i^{(0)}||^2\right\}$$

or

$$C(\mathbf{B}, \mathbf{0}) = \sum_{i=1}^{n_{frm}} \left\{\sum_{k=0}^{3} w_k||\mathbf{B}\bar{\mathbf{r}}_i^{(k)} - \bar{\mathbf{a}}_i^{(k)}||^2\right\} , \tag{18}$$

where

$$w_k = \begin{cases} \alpha & \text{for } k > 0 \\ 1 & \text{for } k = 0 \end{cases}$$

Equation (18) is the same as (3) with $\bar{\mathbf{b}} = \mathbf{0}$ and $m = 4n_{frm}$. Hence it can be solved in closed-form using the same solution.

The reason why we can set $\bar{\mathbf{b}} = \mathbf{0}$, and how to recover $\bar{\mathbf{b}}$ for the original problem, rather than the case when positions are translated to centroidal coordinates, will be shown below.

The fact that $\bar{\mathbf{b}} = \mathbf{0}$ when the original two point sets are centered follows from the direct computation:

$$\frac{\partial}{\partial\bar{\mathbf{b}}}\sum_{i=1}^{n_{frm}} \left\{||\mathbf{B}\bar{\mathbf{r}}_i^{(0)} + \bar{\mathbf{b}} - \bar{\mathbf{a}}_i^{(0)}||^2\right\} = 2\sum_{i=1}^{n_{frm}} \left\{\mathbf{B}\bar{\mathbf{r}}_i^{(0)} + \bar{\mathbf{b}} - \bar{\mathbf{a}}_i^{(0)}\right\} = \mathbf{0} \tag{19}$$

which means

$$\mathbf{B}\left(\sum_{i=1}^{n_{frm}} \bar{\mathbf{r}}_i^{(0)}\right) + n_{frm}\bar{\mathbf{b}} - \left(\sum_{i=1}^{n_{frm}} \bar{\mathbf{a}}_i^{(0)}\right) = \mathbf{0}\,. \tag{20}$$

Since the quantities in parenthesis are zero, $\bar{\mathbf{b}}$ must be zero, as well.

If the optimal value of $\mathbf{B}$, $\mathbf{B}_{opt}$, is computed after centering all positions, the value of the vector $\bar{\mathbf{b}}$ that optimizes the original problem by solving (19) with $\bar{\mathbf{r}}_i$ in place of $\bar{\mathbf{r}}_i^{(0)}$ and likewise for $\bar{\mathbf{a}}_i$. The result is

$$\bar{\mathbf{b}}_{opt} = \bar{\mathbf{a}}_{cm} - \mathbf{B}_{opt}\bar{\mathbf{r}}_{cm}\,, \tag{21}$$

where in this context the subscript "cm" is defined as

$$\bar{\mathbf{x}}_{cm} = \frac{1}{n_{frm}}\sum_{i=1}^{n_{frm}} \bar{\mathbf{x}}_i\,, \tag{22}$$

for any set $\{\bar{\mathbf{x}}_i\}$ consisting of n_{frm} vectors.

B. An $n_{frm} \times 6$ Distance-Jacobian Approach

If $\mathbf{W} \neq \alpha \mathbf{I}$, it is not clear how the closed-form approach to optimal base pose determination presented previously can be used. In addition, if Park's metric, [14], is used in the definition of the cost function, there is no known closed-form expression for the optimal base pose. While gradient descent on $SE(3)$ is one approach to solve the problem, it is not the only solution technique, and one potential alternative is presented here.

Instead of a scalar measure of the goodness of a base pose, consider the following vector of errors:

$$\bar{\mathbf{f}}(\mathbf{b}) = \begin{pmatrix} d^2(\mathbf{b} \circ \mathbf{g}_1, \mathbf{h}_1) \\ \vdots \\ d^2(\mathbf{b} \circ \mathbf{g}_{n_{frm}}, \mathbf{h}_{n_{frm}}) \end{pmatrix} \in \mathbb{R}^{n_{frm}}. \qquad (23)$$

This is a vector error function whose components are the distances between the desired poses and where the manipulator would be able to reach if the base is moved to pose $\mathbf{b}$. By forcing this vector to move toward the zero vector, the resulting value of $\mathbf{b}$ is optimized. Therefore, if we introduce an artificial time parameter and construct a dynamical system

$$-\bar{\mathbf{f}} = \frac{d\bar{\mathbf{f}}}{dt}, \qquad (24)$$

as $t \to \infty$, the desired behavior will be implemented. The way to do this in discrete time is to iteratively search for a small set of motion parameters, δ_1,..., δ_6 such that

$$\frac{d\bar{\mathbf{f}}}{dt} = \mathbf{J}\boldsymbol{\delta}, \qquad (25)$$

where

$$\mathbf{J} = [X_1^R \bar{\mathbf{f}}, ..., X_6^R \bar{\mathbf{f}}].$$

Inverting $\mathbf{J}$ at each timestep to isolate $\boldsymbol{\delta}$, and updating

$$\mathbf{b} \leftarrow \mathbf{b} \circ \exp\left(\sum_{i=1}^{6} \delta_i \mathbf{X}_i\right) \qquad (26)$$

should implement the desired behavior. However, $\mathbf{J}$ is an $n_{frm} \times 6$ matrix, so it can only be inverted in the sense of the overdetermined generalized-inverse.

A similar approach to this was used by Stein to solve the problem of decoding a spherical encoder [15]. Only in that context, spherical motions were considered rather than full rigid-body motions.

C. An "$SE(3)$-Hessian" Approach

Another approach to solving the optimal base placement problem is to expand the cost function $C(\mathbf{b})$ in a multidimensional Taylor series expansion about the specific group element $\mathbf{b}$. The way to do this independently of the parameterization of $SE(3)$ is to replace $\mathbf{b}$ with $\mathbf{b} \circ \exp \sum_{i=1}^{6} \epsilon_i \mathbf{X}_i$ where $|\epsilon_i| << 1$. We can expand

$$C(\mathbf{b} \circ \exp \sum_{i=1}^{6} \epsilon_i \mathbf{X}_i) \approx C(\mathbf{b}) + \bar{\mathbf{a}}^T \bar{\boldsymbol{\epsilon}} + \frac{1}{2} \bar{\boldsymbol{\epsilon}}^T \mathbf{A} \bar{\boldsymbol{\epsilon}}, \qquad (27)$$

where $\bar{\boldsymbol{\epsilon}} = [\epsilon_1, ..., \epsilon_6]^T$, $\bar{\mathbf{a}} = [X_1^R C, ..., X_6^R C]^T$, and

$$\mathbf{A} = \begin{pmatrix} (X_1^R)^2 C & X_2^R(X_1^R C) & \cdots & X_6^R(X_1^R C) \\ X_1^R(X_2^R C) & (X_2^R)^2 C & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ X_1^R(X_6^R C) & \cdots & \cdots & (X_6^R)^2 C \end{pmatrix}.$$

In this paper, $\mathbf{A}$ will be called the *$SE(3)$-Hessian*. Note that $\mathbf{A}$ is not a symmetric matrix because $X_j^R(X_i^R C) \neq X_i^R(X_j^R C)$ due to the noncommutative nature of $SE(3)$. However, it is a general matrix rule that

$$\bar{\boldsymbol{\epsilon}}^T \mathbf{A} \bar{\boldsymbol{\epsilon}} = \bar{\boldsymbol{\epsilon}}^T \mathbf{A}' \bar{\boldsymbol{\epsilon}}, \qquad (28)$$

where

$$\mathbf{A}' = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$$

is symmetric. Therefore, without losing generality, $\mathbf{A}$ can be replaced with $\mathbf{A}'$.

Searching for the best value of $\boldsymbol{\epsilon}$ to perturb $\mathbf{b}$ and reduce the value of C is the same as differentiating Equation (27) with respect to $\boldsymbol{\epsilon}$. The result is

$$\bar{\boldsymbol{\epsilon}} = -(\mathbf{A}')^{-1} \bar{\mathbf{a}}. \qquad (29)$$

Using this value, $\mathbf{b}$ is updated as

$$\mathbf{b} \leftarrow \mathbf{b} \circ \exp \sum_{i=1}^{6} \epsilon_i \mathbf{X}_i \qquad (30)$$

and the process is iterated until $\boldsymbol{\epsilon}$ no longer changes, or at least it is below some prespecified threshold.

D. A $4n_{frm} \times 4$ Generalized-Inverse Approach

Concerning the problem addressed in previous section, another approach to solve the optimal base placement problem is to convert the system of equations $\mathbf{h}_i = \mathbf{b} \circ \mathbf{g}_i$ for $i = 1, ..., n_{frm}$ to the form $\mathbf{b} = \mathbf{h}_i \circ \mathbf{g}_i^{-1}$. It is an overconstrained problem to find one $\mathbf{b} \in SE(3)$ to simultaneously solve all of these equations. We can write this as a system

$$\mathbf{C} = \begin{pmatrix} \mathbf{h}_1 \circ \mathbf{g}_1^{-1} \\ \vdots \\ \mathbf{h}_{n_{frm}} \circ \mathbf{g}_{n_{frm}}^{-1} \end{pmatrix} = \mathbf{J}\mathbf{b}, \qquad (31)$$

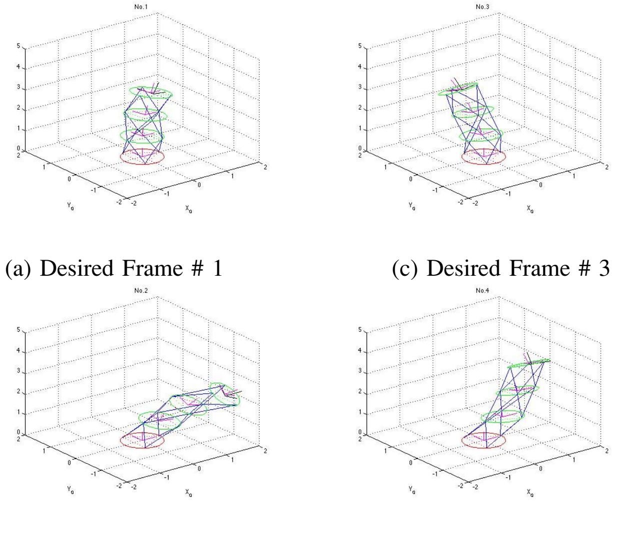
where $\mathbf{C} \in \mathbb{R}^{4n_{frm} \times 4}$ and

$$\mathbf{J} = \begin{pmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{pmatrix} \in \mathbb{R}^{4n_{frm} \times 4},$$

where $\mathbf{I}$ is the 4×4 identity.

We can solve this system using a weighted generalized inverse as

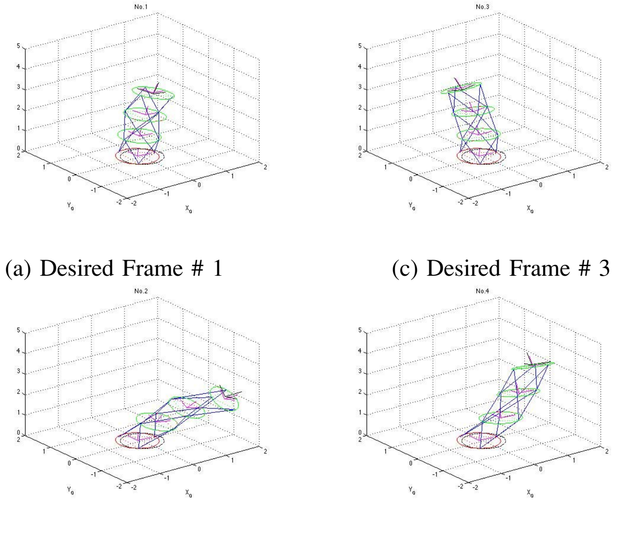
$$\begin{aligned} \mathbf{b} &= \mathbf{J}^{\dagger} \mathbf{C} \\ &= (\mathbf{J}^T \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{W} \mathbf{C}, \end{aligned} \qquad (32)$$



(a) Desired Frame # 1 (c) Desired Frame # 3

(b) Desired Frame # 2 (d) Desired Frame # 4

Fig. 3. Baseline Design Example (Generalized-Inverse Approach)



(a) Desired Frame # 1 (c) Desired Frame # 3

(b) Desired Frame # 2 (d) Desired Frame # 4

Fig. 4. Synthesis Result Example (Generalized-Inverse Approach)

where $\mathbf{W}$ is called a "weight matrix" and is a symmetric positive definite matrix. $\mathbf{W}$ is often taken as a block-diagonal matrix consisting of n_{frm} copies of the matrix

$$\mathbf{W}' = \begin{pmatrix} \alpha\mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{33}$$

on its diagonal. Here α is a length parameter to reconcile translational and orientational quantities.

A series of simulations is conducted to confirm the feasibility of these approaches shown above. Fig. 3 and 4 describe an example case for the Generalized-Inverse approach. Note that the arm architecture is not altered in the computation. The original base plate, at the identity frame of reference, is also depicted in the figures of synthesis results. The proposed kinematic computation method successfully obtained the base placement frame, which is similar to the one obtained with the other approaches (no figures).

References

[1] I. Ebert-Uphoff, "On the development of Discretely-Actuated Hybrid-Serial-Parallel manipulators," Ph.D. Dissertation, Johns Hopkins University, Department of Mechanical Engineering, Baltimore, MD, May 1997.

[2] K. Miyahara and G. S. Chirikjian, "Discretely actuated robotic manipulator (d-arm) for mems application," in *Proceedings of International Conference on Technology Fusion (ICTF 2006)*. Seoul, Korea: Artificial Muscle Research Center (AMRC), Konkuk University, May 2006.

[3] V. Sujan and S. Dubowsky, "Design of a lightweight hyper-redundant deployable binary manipulator," *Journal of Mechanical Design, Transaction of ASME*, vol. 126, pp. 29–39, 2004.

[4] G. Chirikjian, "Kinematic synthesis of mechanisms and robotic manipulators with binary actuators," *Journal of Mechanical Design, Transaction of ASME*, vol. 117, pp. 573–580, Dec. 1995.

[5] ——, "A binary paradigm for robotic manipulators," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. San Diego, CA: IEEE R&A, May 1994, pp. 3063–3069.

[6] G. Chirikjian and I. Ebert-Uphoff, "Numerical convolution on the Euclidean group with applications to workspace generation," *IEEE Transaction on Robotics and Automation*, vol. 14, no. 1, pp. 123–136, Feb. 1998.

[7] G. Chirikjian and A. Kyatkin, *Engineering Applications of Noncommutative Harmonic Analysis*. Boca Raton, FL: CRC Press, 2000.

[8] J. Selig, "Three problems in robotics," in *Proceedings of the Ball Symposium*, Cambridge, MA, July 9-11 2000.

[9] K. Kanatani, "Analysis of 3-d rotation fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 543–549, May 1994.

[10] J. MacKenzie, "The estimation of an orientation relationship," *Acta Crystallographica*, vol. 10, pp. 61–62, 1957.

[11] F. Markley, "Attitude determination using vector observations and the singular value decomposition," *Journal of the Astronautical Sciences*, vol. 36, no. 3, pp. 245–258, 1988.

[12] A. Nádas, "Least squares and maximum likelihood estimates of rigid motion," *IBM Research Report RC 6945*, no. 29783, Jan. 17 1978.

[13] G. Chirikjian and S. Zhou, "Metrics on motion and deformation of solid models," *Journal of Mechanical Design, Transaction of ASME*, vol. 120, pp. 252–261, 1998.

[14] F. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *Journal of Mechanical Design, Transactions of the ASME*, vol. 117, pp. 48–54, Mar. 1995.

[15] D. Stein, "Design of a spherical stepper motor system," Ph.D. Dissertation, Johns Hopkins University, Department of Mechanical Engineering, Baltimore, MD, Aug. 2001.